#include <iostream>

#include <fstream>

#include <vector>

#include <unordered\_map>

#include <sstream>

// Structure to store a macro definition in MDT

struct MDTEntry {

std::string line;

};

// Structure for storing information in MNT

struct MNTEntry {

std::string macroName;

int mdtIndex; // Points to the start of the macro definition in MDT

};

// Macro Processor class

class MacroProcessor {

private:

std::unordered\_map<std::string, MNTEntry> MNT; // Macro Name Table

std::vector<MDTEntry> MDT; // Macro Definition Table

int mdtIndex;

public:

MacroProcessor() : mdtIndex(0) {}

// Process a line to check for macro definitions

void processLine(const std::string& line, std::ofstream& intermediateFile, std::vector<std::string>::iterator& it, std::vector<std::string>::iterator end) {

std::istringstream stream(line);

std::string word;

stream >> word;

if (word == "MACRO") {

defineMacro(it, end);

} else if (MNT.find(word) != MNT.end()) {

intermediateFile << "CALL " << word << "\n";

} else {

intermediateFile << line << "\n";

}

}

// Method to define a macro

void defineMacro(std::vector<std::string>::iterator& it, std::vector<std::string>::iterator end) {

std::istringstream stream(\*it);

std::string macroName, param;

stream >> macroName >> macroName; // skip "MACRO" and read macro name

MNTEntry entry;

entry.macroName = macroName;

entry.mdtIndex = mdtIndex;

// Add macro to MNT

MNT[macroName] = entry;

// Add macro header to MDT

MDT.push\_back({macroName + " " + param});

mdtIndex++;

// Read macro body lines

++it; // Move to the next line after "MACRO"

while (it != end) {

std::istringstream bodyStream(\*it);

std::string firstWord;

bodyStream >> firstWord;

if (firstWord == "MEND") {

MDT.push\_back({"MEND"});

mdtIndex++;

break;

} else {

MDT.push\_back({\*it});

mdtIndex++;

}

++it;

}

}

// Display MNT and MDT tables

void displayTables() {

std::cout << "\nMacro Name Table (MNT):\n";

for (const auto& entry : MNT) {

std::cout << "Macro: " << entry.second.macroName

<< ", MDT Index: " << entry.second.mdtIndex << "\n";

}

std::cout << "\nMacro Definition Table (MDT):\n";

for (int i = 0; i < MDT.size(); i++) {

std::cout << i << ": " << MDT[i].line << "\n";

}

}

};

int main() {

MacroProcessor processor;

std::ofstream intermediateFile("intermediate\_code.txt");

// Sample source code with macros

std::vector<std::string> sourceCode = {

"MACRO ADDM &ARG1, &ARG2",

"LDA &ARG1",

"ADD &ARG2",

"STA &ARG1",

"MEND",

"START",

"MOVEM A, B",

"ADDM NUM1, NUM2",

"END"

};

// Process each line

for (auto it = sourceCode.begin(); it != sourceCode.end(); ++it) {

processor.processLine(\*it, intermediateFile, it, sourceCode.end());

}

intermediateFile.close();

// Display tables

processor.displayTables();

return 0;

}